Teaching Computers To See From Space: Deep Learning and Sentinel 2

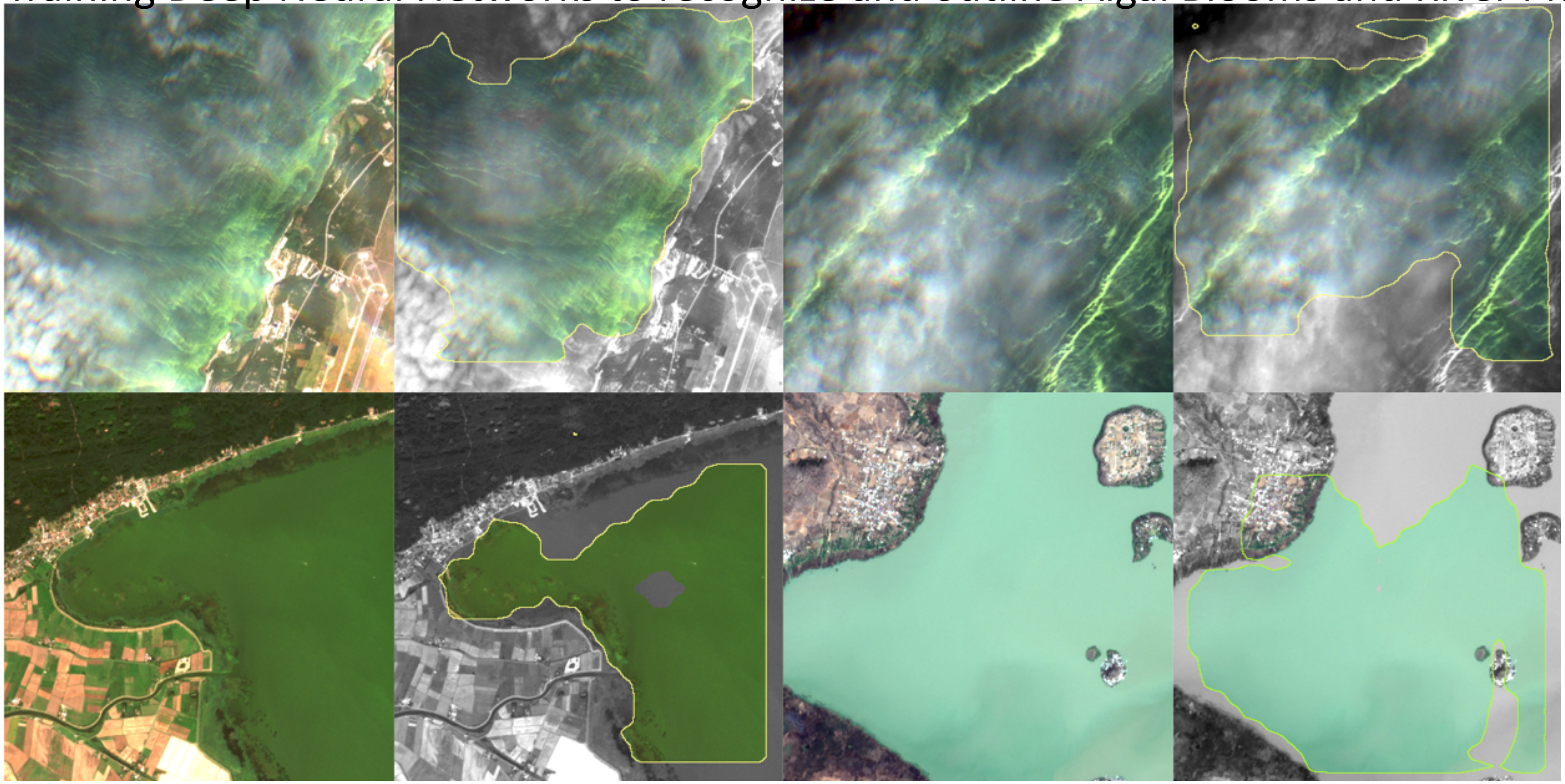
Stephen Goult

Stefan Simis, Chunbo Luo, Shubha Sathyendranath

PML Plymouth Marine Laboratory
esa European Space Agency
dragon 4 cooperation ESA–MOST China programme
UNIVERSITY OF EXETER

What are you doing?

Training Deep Neural Networks to recognize and outline Algal Blooms and River Plumes



What is Deep Learning?

Neural networks that use a cascade of layers to build feature descriptions

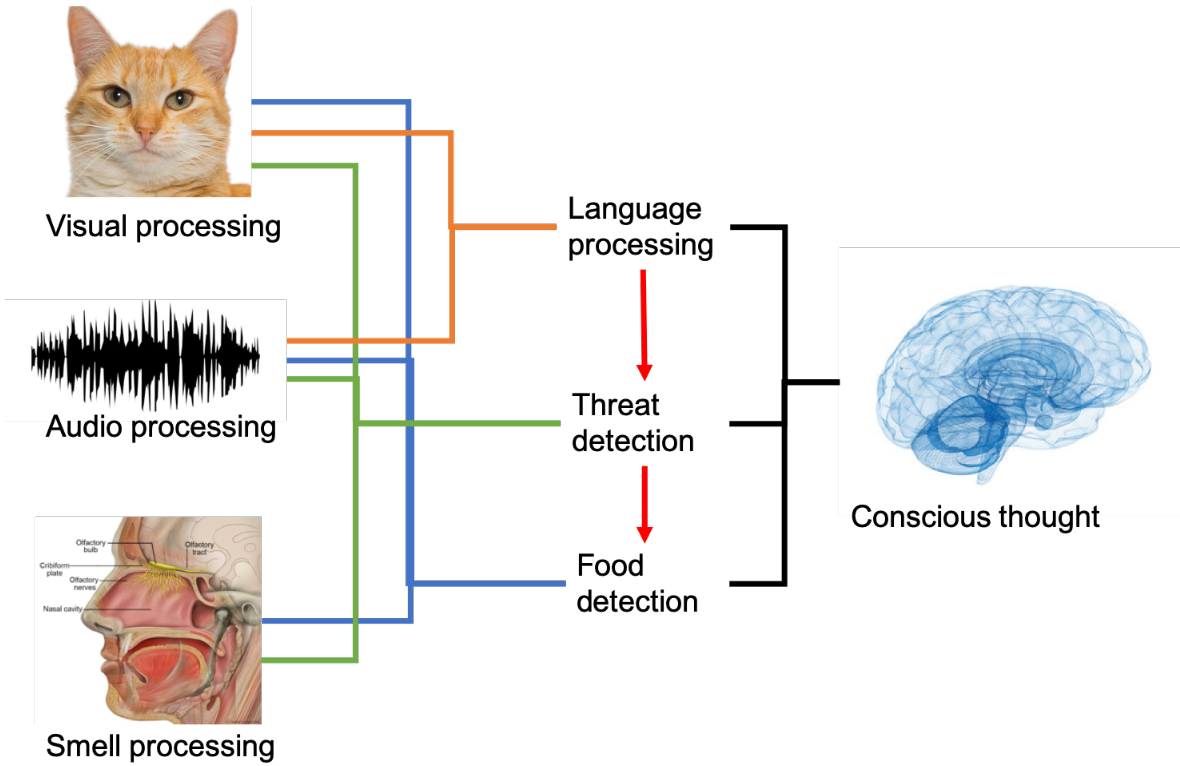
Build an abstract representation of an output, which is used to match features to classification (or other outputs)

Covers a broad range of applications, from image classification to noise cancelling and even data generation

So its Artificial Intelligence?

We can think of these models as agents which can feed into each other.

Here is a bad analogy:



Object description

We know it’s a cat because it looks like one

We know it’s a cat because it has cat ears

We know they are ears because they are on top of a head shaped object

We know they’re cat ears because they are pointed

Of course we also know what it is because of the way it moves, sounds smells etc…

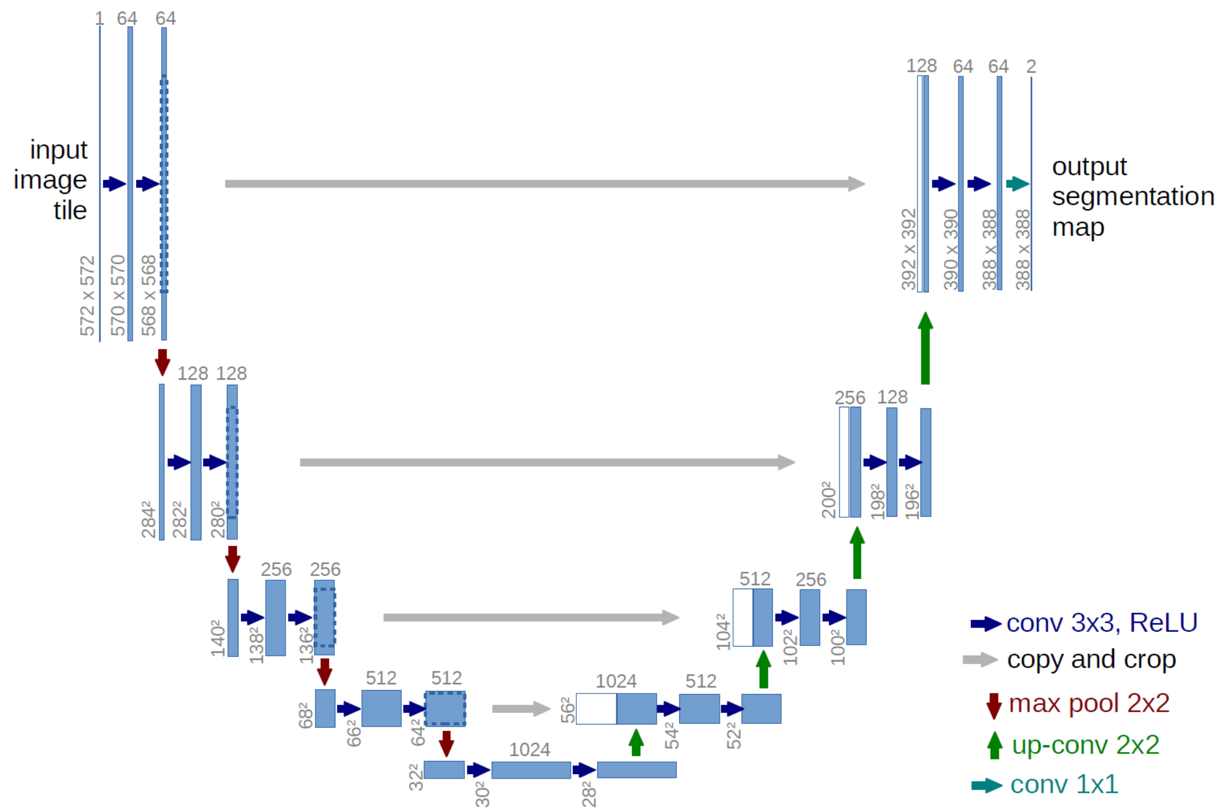
Instance segmentation and classification

Given all the descriptors in my data, tell me:

- What it is
- Where it is
- What area it occupies
- What else is around it
- How confident you are about all of these factors

All of this is done with a Convolutional Neural Network

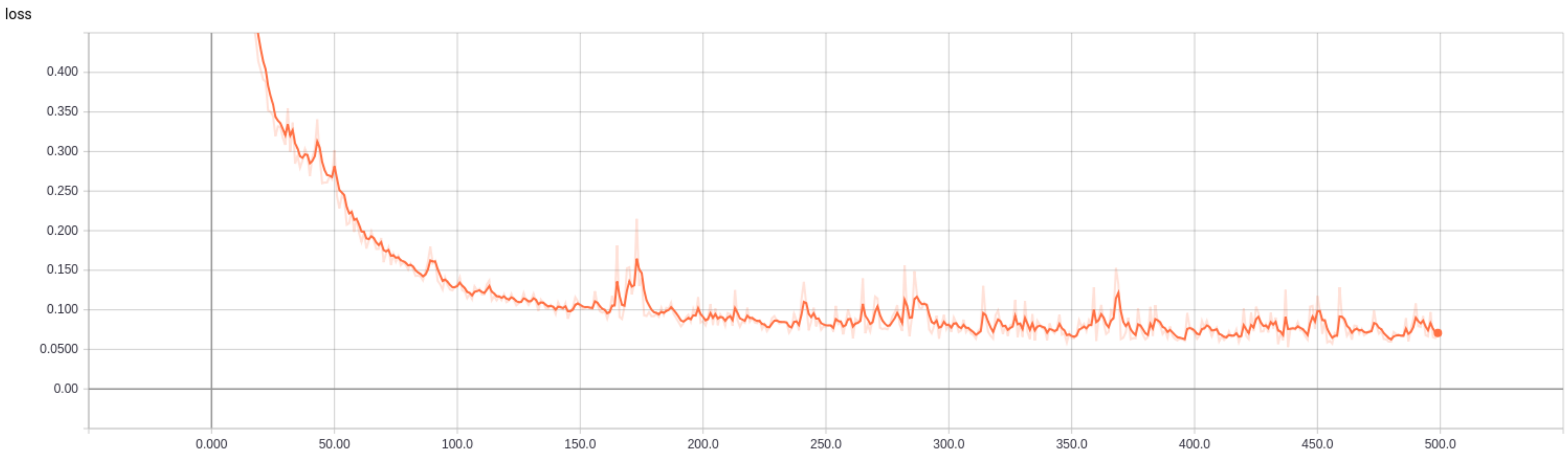
Convolutional Neural Network (CNN)



This depiction is for an implementation called U-Net, designed for medical imaging segmentation

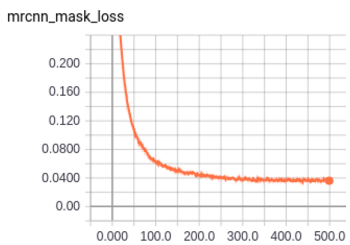
For this work, Mask R-CNN is the current model selected. Its architecture is less easy to depict, but Feature Hierarchical Extraction works in a similar way.

Loss

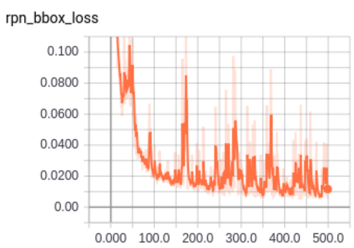


loss
0.400
0.350
0.300
0.250
0.200
0.150
0.100
0.0500
0.00
0.000
50.00
100.0
150.0
200.0
250.0
300.0
350.0
400.0
450.0
500.0

mrcnn_class_loss
0.0400
0.0300
0.0200
0.0100
0.00
0.000 100.0 200.0 300.0 400.0 500.0



mrcnn_mask_loss
0.200
0.160
0.120
0.0800
0.0400
0.00
0.000 100.0 200.0 300.0 400.0 500.0

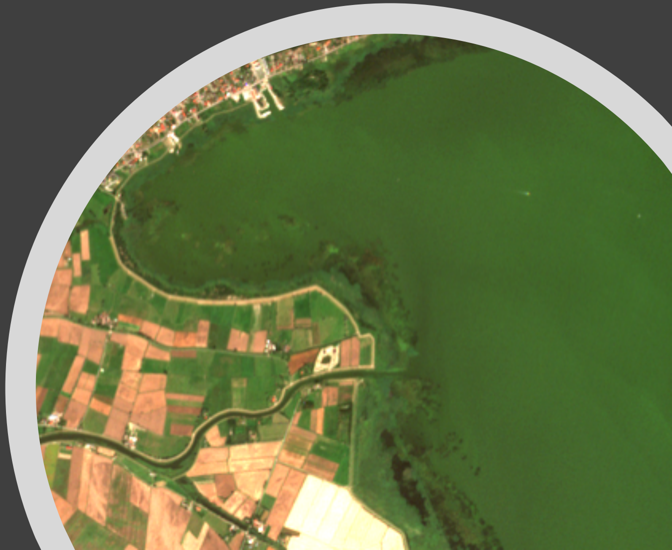


rpn_bbox_loss
0.100
0.0800
0.0600
0.0400
0.0200
0.00
0.000 100.0 200.0 300.0 400.0 500.0

n_class_loss
5.000e-3
4.000e-3
3.000e-3
2.000e-3
1.000e-3
0.00
0.000 100.0 200.0 300.0 400.0 500.0

Shortcuts to results

- Training Deep Neural Networks from scratch takes time and huge amounts of data
  - > 1,000,000 images
  - Weeks and even months on a single GPU
- For many state-of-the-art architectures, the code and training weights have already been made available so we can use a technique called “transfer learning”
  - This means the feature descriptors that have already been learnt can be reassembled to match our new object, this is effective with satellite imagery

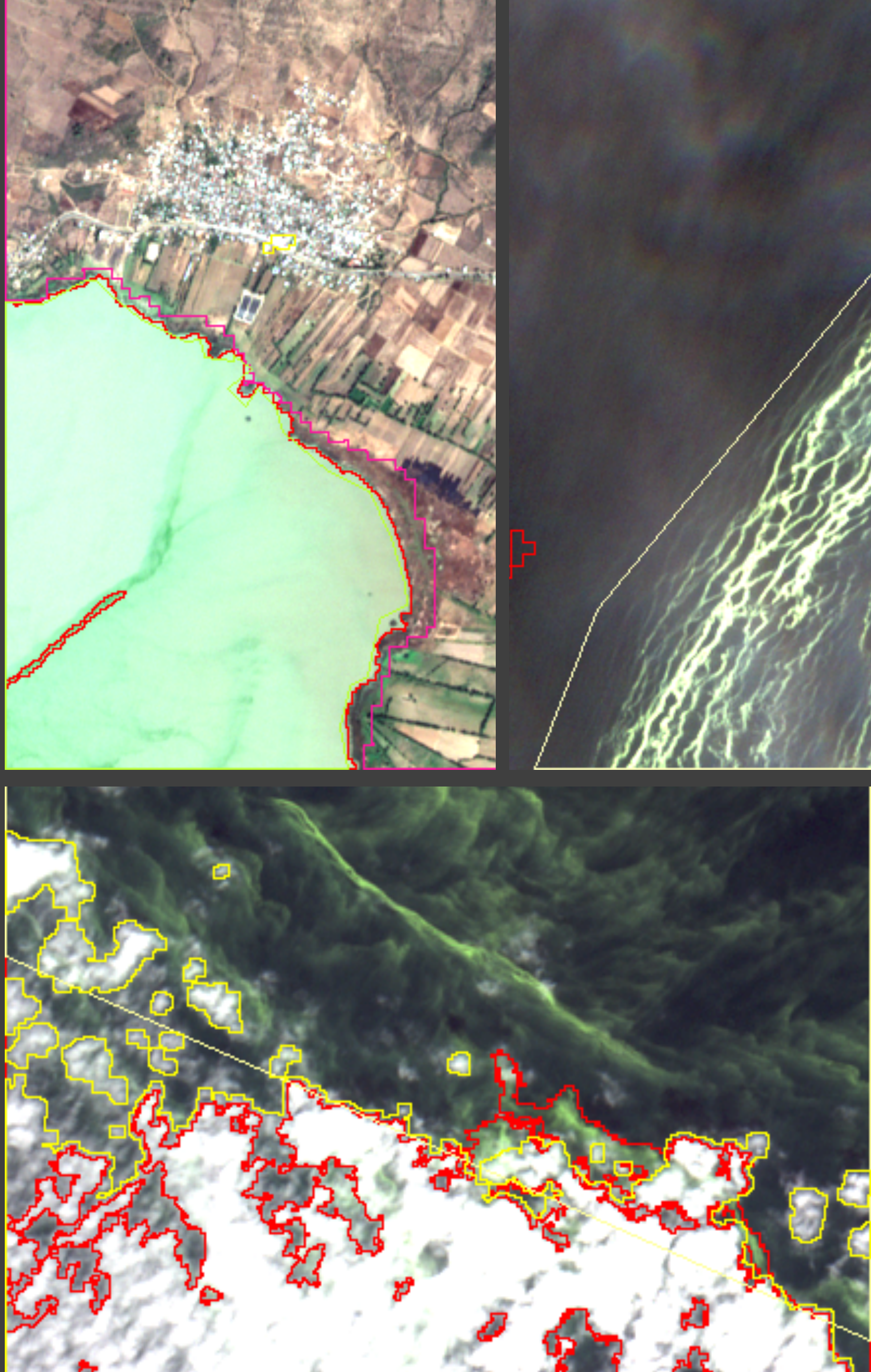


First attempt

Ran it on a CPU	• Took 3 weeks to train • Locked up the whole computer
Trained with 10 2000 x 2000 pixel tiles sliced from Sentinel 2	• The inputs were rubbish, so the outputs were worse
Attempted transfer learning, actually randomized my own weights	• Classified the entire image as any object requested

Expanding the training set

- Created a training dataset with over 5000 images tiled from 101 sentinel 2 scenes with confirmed cases of Cyanobacteria present
    - 400 x 400 images
- To make it broadly usable scenes are processed with both ACOLITE and POLYMER atmospheric correctors, and stored all of the variables they produce
    - Select the bands needed for a model
- This is being expanded when time allows



Expanding the hardware

- Brought a Graphics Processing Unit (GPU) from home
  - Can process 1 image at a time
  - 25 images a second
- Added 10 Tb of raid array with 1 Tb of data cache on the processing node
- Added 200 Gigabytes of swap file
- Steve, Dan, Pete and myself asked NERC for £1 million in funding buy a GPU High Performance Cluster (HPC) and got it!
  - ~40 GPUs
  - Can be networked together
  - Can process 120 images at a time at 100%
  - 3000 images a second

Second attempt

Trained with RGB composites of the data

- Most in line with the current weights available

Utilised a GPU this time

- 6 hours –> 6 minutes per epoch because of the GPU

Trained using a subset of the available images

- 1 day to run on ~1000 images
- 55% of pixels were correctly classified

Band configurations

	443	492	560	665	704	740	783	833	865	1614	2202
RGB		X	X	X							
RGB + SWIR		X	X	X						X	X
All Bands	X	X	X	X	X	X	X	X	X	X	X

Values scaled by 256 to match image data weights

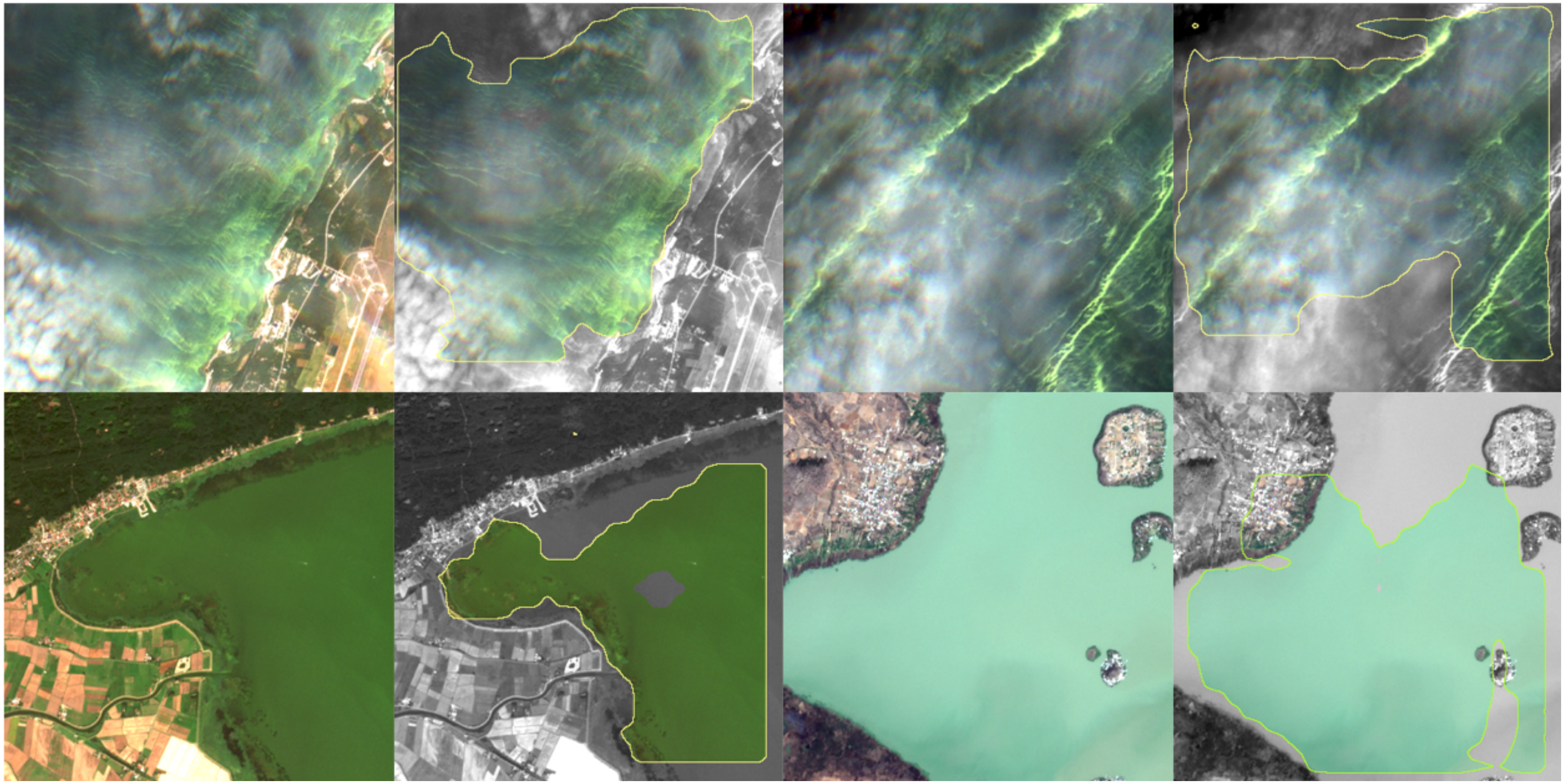
Results

CONFIG:	RGB BANDS	ALL TOP OF ATMOSPHERE	RGB + SWIR	RALEIGH RGB	ALL RALEIGH CORRECTED BANDS	RALEIGH RGB + SWIR
Mean Accuracy (% pixels correct)	72.5	73.7	77.263	71.83	67.7	77.754
Mean Confusion as Cloud	3.25	2.3	0.784	18.51	7.17	21.06
Mean Confusion as land	15.86	7.5	9.35	9.57	14.47	9.49
Mean Confusion as Water	34.48	38.5	16.35	2.742	13.35	9.46

*Polymer not included as results are still being generated

Results

Using RGB + SWIR band configuration



Discussion

For simple classifications (ie, yes there is a bloom here/no there isn't) we don't actually necessarily need the long, complex data pre processing

Variation in the training dataset is key, and an overabundance of certain scenarios will over fit to those environments

With more hardware the processing can be sped up and improved with larger image sizes

Plan for the next year

Extend	Extend the models I already have with more data on the GPU HPC
Expand	Expand classifications beyond a binary statement of water contents into something more meaningful
Evaluate	Evaluate the sensitivity of hierarchical features in larger images • E.g. 1000 x 1000
Trace	Finally get to work on the tracing/combination of observations over time